Customizing an Information Extraction System to a New Domain

Mihai Surdeanu, David McClosky, Mason R. Smith, Andrey Gusev,
and Christopher D. Manning
Department of Computer Science
Stanford University
Stanford, CA 94305
{mihais,mcclosky,mrsmith,manning}@stanford.edu
agusev@cs.stanford.edu

Abstract

We introduce several ideas that improve the performance of supervised information extraction systems with a pipeline architecture, when they are customized for new domains. We show that: (a) a combination of a sequence tagger with a rule-based approach for entity mention extraction yields better performance for both entity and relation mention extraction; (b) improving the identification of syntactic heads of entity mentions helps relation extraction; and (c) a deterministic inference engine captures some of the joint domain structure, even when introduced as a post-processing step to a pipeline system. All in all, our contributions yield a 20% relative increase in F1 score in a domain significantly different from the domains used during the development of our information extraction system.

1 Introduction

Information extraction (IE) systems generally consist of multiple interdependent components, e.g., entity mentions predicted by an entity mention detection (EMD) model connected by relations via a relation mention detection (RMD) component (Yao et al., 2010; Roth and Yih, 2007; Surdeanu and Ciaramita, 2007). Figure 1 shows a sentence from a sports domain where both entity and relation mentions are annotated. When training data exists, the best performance in IE is generally obtained by supervised machine learning approaches. In this scenario, the typical approach for domain customization is apparently straightforward: simply retrain on data from the new domain (and potentially tune model parameters). In this paper we argue that, even when considerable training data is available, this is not sufficient to maximize performance. We apply several simple ideas that yield a significant performance boost, and can be implemented with minimal effort. In particular:

- We show that a combination of a conditional random field model (Lafferty et al., 2001) with a rule-based approach that is recall oriented yields better performance for EMD and for the downstream RMD component. The rule-based approach includes gazetteers, which have been shown to be important by Mikheev et al. (1999), among others.
- We improve the unification of the predicted semantic annotations with the syntactic analysis of the corresponding text, i.e., finding the syntactic head of a given semantic constituent. Since many features in an IE system depend on syntactic analysis, this leads to more consistent features and better extraction models.
- We add a simple inference engine that generates additional relation mentions based solely on the relation mentions extracted by the RMD model. This engine mitigates some of the limitations of a text-based RMD model, which cannot extract relations not explicitly stated in text.

We investigate these ideas using an IE system that performs recognition of entity mentions followed by extraction of binary relations between these mentions. We used as target a sports domain that is significantly different from the corpora previously used with this IE system. The target domain is also significantly different from the dataset used to train the

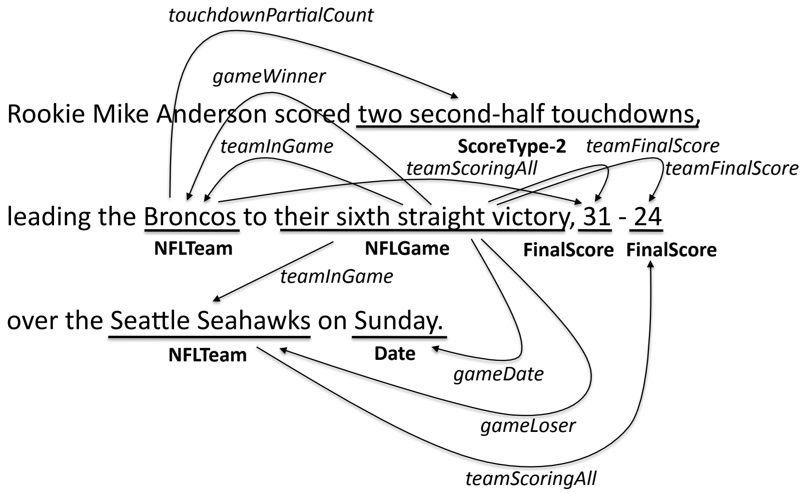


Figure 1: Sample sentence from the NFL domain. The domain contains entity mentions (underlined with entity types in bold) and binary relations between entity mentions (indicated by arrows; relation types are italicized).

supporting natural language processing tools (e.g., syntactic parser). Our investigation shows that, despite their simplicity, all our proposals help, yielding a 20% relative improvement in RMD F1 score.

The paper is organized as follows: Section 2 surveys related work. Section 3 describes the IE system used. We cover the target domain that serves as use case in this paper in Section 4. Section 5 introduces our ideas and evaluates their impact in the target domain. Finally, Section 6 concludes the paper.

2 Related Work

Other recent works have analyzed the robustness of information extraction systems. For example, Florian et al. (2010) observed that EMD systems perform badly on noisy inputs, e.g., automatic speech transcripts, and propose system combination (similar to our first proposal) to increase robustness in such scenarios. Ratinov and Roth (2009) also investigate design challenges for named entity recognition, and showed that other design choices, such as the representation of output labels and using features built on external knowledge, are more important than the learning model itself. These works are conceptually similar to our paper, but we propose several additional directions to improve robustness, and we investigate their impact in a complete IE system instead of just EMD.

Several of our lessons are drawn from the BioCreative challenge[1] and the BioNLP shared task (Kim et al., 2009). These tasks have shown the importance of high quality syntactic annotations and using heuristic fixes to correct systematic errors (Schuman and Bergler, 2006; Poon and Vanderwende, 2010, among others). Systems in the latter task have also shown the importance of high recall in the earlier stages of pipeline system.

3 Description of the Generic IE System

We illustrate our proposed ideas using a simple IE system that implements a pipeline architecture: entity mention extraction followed by relation mention extraction. Note however that the domain customization discussion in Section 5 is independent of the system architecture or classifiers used for EMD and RMD, and we expect the proposed ideas to apply to other IE approaches as well.

We performed all pre-processing (tokenization, part-of-speech (POS) tagging) with the Stanford CoreNLP toolkit.[2] For EMD we used the Stanford named entity recognizer (Finkel et al., 2005). In all our experiments we used a generic set of features (“macro”) and the IO notation[3] for entity mention labels (e.g., the labels for the tokens “over the Seattle Seahawks on Sunday” (from Figure 1) are encoded as “O O NFLTEAM NFLTEAM O DATE”).

[1] http://biocreative.sourceforge.net/

[2] http://nlp.stanford.edu/software/corenlp.shtml

[3] The IO notation facilitates faster inference than the IOB or IOB2 notations with minimal impact on performance, when there are fewer adjacent mentions with the same type.

Argument Features	– Head words of the two arguments and their combination – Entity mention labels of the two arguments and their combination
Syntactic Features	– Sequence of dependency labels in the dependency path linking the heads of the two arguments – Lemmas of all words in the dependency path – Syntactic path in the constituent parse tree between the largest constituents headed by the same words as the two arguments (similar to Gildea and Jurafsky (2002))
Surface Features	– Concatenation of POS tags between arguments – Binary indicators set to true if there is an entity mention with a given type between the two arguments

Table 1: Feature set used for RMD.

The RMD model was built from scratch as a multi-class classifier that extracts binary relations between entity mentions in the same sentence. During training, known relation mentions become positive examples for the corresponding label and all other possible combinations between entity mentions in the same sentence become negative examples. We used a multiclass logistic regression classifier with L2 regularization. Our feature set is taken from (Yao et al., 2010; Mintz et al., 2009; Roth and Yih, 2007; Surdeanu and Ciaramita, 2007) and models the relation arguments, the surface distance between the relation arguments, and the syntactic path between the two arguments, using both constituency and dependency representations. For syntactic information, we used the Stanford parser (Klein and Manning, 2003) and the Stanford dependency representation (de Marneffe et al., 2006).

For RMD, we implemented an additive feature selection algorithm similar to the one in (Surdeanu et al., 2008), which iteratively adds the feature with the highest improvement in F1 score to the current feature set, until no improvement is seen. The algorithm was configured to select features that yielded the best combined performance on the dataset from Roth and Yih (2007) and the training partition of ACE 2007.[4] We used ten-fold cross validation on both datasets. We decided to use a standard F1 score to evaluate RMD performance rather than the more complex ACE score because we believe that the former is more interpretable. We used gold entity mentions for the feature selection process. Table 1 summarizes the final set of features selected.

Documents	Words	Entity Mentions	Relation Mentions
110	70,119	2,188	1,629

Table 2: Summary statistics of the NFL corpus, after our conversion to binary relations.

Despite its simplicity, our approach achieves comparable performance with other state-of-the-art results reported on these datasets (Roth and Yih, 2007; Surdeanu and Ciaramita, 2007). For example, Surdeanu and Ciaramita report a RMD F1 score of 59.4 for ACE relation types (i.e., ignoring subtypes) when gold entity mentions are used. Under the same conditions, our RMD model obtains a F1 score of 59.2.

4 Description of the Target Domain

In this paper we report results on the "Machine Reading NFL Scoring" corpus.[5] This corpus was developed by LDC for the DARPA Machine Reading project. The corpus contains 110 newswire articles on National Football League (NFL) games. The annotations cover game information, such as participating teams, winners and losers, partial (e.g., a single touchdown or three field goals) and final scores. Most of the annotated relations in the original corpus are binary (e.g. GAMEDATE(NFLGAME, DATE)) but some are n-ary relations or include other attributes in addition of the relation type. We reduce these to annotations compatible with our RMD approach as follows:

- We concatenate the cardinality of each scoring event (i.e. how many scoring events are being talked about) to the corresponding SCORETYPE entity label. Thus SCORETYPE-2 indicates that there were two of a given type of scoring event (touchdown, field goal, etc.). This operation is necessary because the cardinality of scoring events is originally annotated as an additional attribute of the SCORETYPE

[4] LDC catalog numbers LDC2006E54 and LDC2007E11

[5] LDC catalog number LDC2009E112

Entity Mentions	Correct	Predicted	Actual	P	R	F1
Date	141	190	174	74.2	81.0	77.5
FinalScore	299	328	347	91.2	86.2	88.6
NFLGame	71	109	147	65.1	48.3	55.5
NFLPlayoffGame	8	25	38	32.0	21.1	25.4
NFLTeam	651	836	818	77.9	79.6	78.7
ScoreType-1	329	479	525	68.7	62.7	65.5
ScoreType-2	49	68	79	72.1	62.0	66.7
ScoreType-3	17	26	36	65.4	47.2	54.8
ScoreType-4	6	11	14	54.5	42.9	48.0
Total	1571	2076	2188	75.7	71.8	73.7

Relation Mentions	Correct	Predicted	Actual	P	R	F1
fieldGoalPartialCount	33	41	101	80.5	32.7	46.5
gameDate	32	36	115	88.9	27.8	42.4
gameLoser	22	44	124	50.0	17.7	26.2
gameWinner	6	15	123	40.0	4.9	8.7
teamFinalScore	95	101	232	94.1	40.9	57.1
teamInGame	49	105	257	46.7	19.1	27.1
teamScoringAll	202	232	321	87.1	62.9	73.1
touchDownPartialCount	156	191	322	81.7	48.4	60.8
Total	595	766	1629	77.7	36.5	49.7

Table 3: Baseline results: stock system without any domain customization. Correct/Predicted/Actual indicate the number of mentions (entities or relations) that are correctly predicted/predicted/gold. P/R/F1 indicate precision/recall/F1 scores for the corresponding label.

entity and our EMD approach does not model mention attributes.

- We split all n-ary relations into several new binary relations. For example, the original teamFinalScore(NFLTeam, NFLGame, FinalScore) relation is split into three binary relations: teamScoringAll(NFLTeam, FinalScore), teamInGame(NFLGame, NFLTeam), and teamFinalScore(NFLGame, FinalScore).

Figure 1 shows an example annotated sentence after the above conversion and Table 2 lists the corpus summary statistics for the new binary relations.

The purpose behind this corpus is to encourage the development of systems that answer structured queries that go beyond the functionality of information retrieval engines, e.g.:

> "For each NFL game, identify the winning and losing teams and each team's final score in the game."
>
> "For each team losing to the Green Bay Packers, tell us the losing team and the number of points they scored."[6]

5 Domain Customization

Table 3 lists the results of the generic IE system described in Section 3 on the NFL domain. Throughout this paper we will report results using ten-fold cross-validation on all 110 documents in the corpus.[7] We consider an entity mention as correct if both its boundaries and label match exactly the gold mention. We consider a relation mention correct if both its arguments and label match the gold relation mention. For RMD, we report results using the actual mentions predicted by our EMD model (instead of using gold entity mentions for RMD). For clarity, we do not show in the tables some labels that are highly uncommon in the data (e.g., ScoreType-5 appears only four times in the entire corpus); but the "Total" results include all entity and relation mentions.

Table 3 shows that the stock IE system obtains an

[6] These queries would be written in a formal language but are presented here in English for clarity.

[7] Generally, we do not condone reporting results using cross-validation because it may be a recipe for over-fitting on the corresponding corpus. However, all our domain customization ideas were developed using outside world and domain knowledge and were not tuned on this data, so we believe that there is minimal over-fitting in this case.

Entity Mentions	P	R	F1
Date	74.2	81.0	77.5
FinalScore	91.3	87.3	89.2
NFLGame	61.2	48.3	54.0
NFLPlayoffGame	33.3	21.1	25.8
NFLTeam	77.9	81.3	79.5
ScoreType-1	68.8	62.3	65.4
ScoreType-2	72.1	62.0	66.7
ScoreType-3	65.4	47.2	54.8
ScoreType-4	54.5	42.9	48.0
Total	75.6	72.5	74.0

Relation Mentions	P	R	F1
fieldGoalPartialCount	78.0	31.7	45.1
gameDate	91.4	27.8	42.7
gameLoser	50.0	18.5	27.1
gameWinner	40.0	4.9	8.7
teamFinalScore	94.1	40.9	57.1
teamInGame	45.9	19.5	27.3
teamScoringAll	87.0	64.8	74.3
touchDownPartialCount	82.4	49.4	61.7
Total	77.6	37.1	50.2

Table 4: Performance after gazetteer-based features were added to the EMD model.

Entity Mentions	P	R	F1
Date	74.2	81.0	77.5
FinalScore	91.3	87.3	89.2
NFLGame	61.2	48.3	54.0
NFLPlayoffGame	33.3	21.1	25.8
NFLTeam	71.4	96.9	82.3
ScoreType-1	68.8	62.3	65.4
ScoreType-2	72.1	62.0	66.7
ScoreType-3	65.4	47.2	54.8
ScoreType-4	54.5	42.9	48.0
Total	72.8	78.4	75.5

Relation Mentions	P	R	F1
fieldGoalPartialCount	81.2	38.6	52.3
gameDate	93.9	27.0	41.9
gameLoser	51.1	19.4	28.1
gameWinner	38.9	5.7	9.9
teamFinalScore	94.1	40.9	57.1
teamInGame	47.4	24.5	32.3
teamScoringAll	87.0	68.8	76.9
touchDownPartialCount	81.6	56.5	66.8
Total	77.2	40.6	53.2

Table 5: Performance after gazetteer-based features were added to the EMD model, and NFLTeam entity mentions were extracted using the rule-based model rather than classification.

EMD F1 score of 73.7 and a RMD F1 score of 49.7. These are respectable results, in line with state-of-the-art results in other domains.[8] However, there are some obvious areas for improvement. For example, the score for a few relations (e.g., GAMELOSER and GAMEWINNER) is quite low. This is caused by the fact that these relations are often not explicitly stated in text but rather implied (e.g., based on team scores). Furthermore, the low recall of entity types that are crucial for all relations (e.g., NFLTEAM and NFLGAME) negatively impacts the overall recall of RMD.

5.1 Combining a Rule-based Model with Conditional Random Fields for EMD

A straightforward way to improve EMD performance is to construct domain-specific gazetteers and include gazetteer-based features in the model. We constructed a NFL-specific gazetteer as follows: (a) we included all 32 NFL team names; (b) we built a lexicon for NFLGame nouns and verbs that included game types (e.g., "semi-final", "quarter-final") and typical game descriptors. The game descriptors were manually bootstrapped from three seed words ("victory", "loss", "game") using Dekang Lin's dependency-based thesaurus.[9] This process added other relevant game descriptors such as "triumph", "defeat", etc. All in all, our gazetteer includes 32 team names and 50 game descriptors. The gazetteer was built in less than four person hours.

We added features to our EMD model to indicate if a sequence of words matches a gazetteer entry, allowing approximate matches (e.g., "Cowboys" matches "Dallas Cowboys"). Table 4 lists the results after this change. The improvements are modest: 0.3 for both EMD and RMD, caused by a 0.8 improvement for NFLTEAM. The score for NFLGAME suffers a loss of 1.5 F1 points, probably caused by the fact that our NFLGAME gazetteer is incomplete.

These results are somewhat disappointing: even though our gazetteer contains an exhaustive list of NFL team names, the EMD recall for NFLTEAM is still relatively low. This happens because city

[8] As a comparison, the best RMD system in ACE 2007 obtained an ACE score of less than 35%, even though the ACE score gives credit for approximate matches of entity mention boundaries (Surdeanu and Ciaramita, 2007).

[9] `http://webdocs.cs.ualberta.ca/~lindek/Downloads/sim.tgz`

names that are not references to team names are relatively common in this corpus, and the CRF model favors the generic city name interpretation. However, since the goal is to answer structured queries over the extracted relations, we would prefer a model that favors recall for EMD, to avoid losing candidates for RMD. While this can be achieved in different ways (Minkov et al., 2006), in this paper we implement a very simple approach: we recognize NFLTeam mentions with a rule-based system that extracts all token sequences that begin, end, or are equal to a known team name. For example, "Green Bay" and "Packers" are marked as team mentions, but not "Bay". Note that this approach is prone to introducing false positives, e.g., "Green" in the above example. For all other entity types we use the CRF model with gazetteer-based features. Table 5 lists the results for this model combination. The table shows that the RMD performance is improved by 3 F1 points. The F1 score for NFLTeam mentions is also improved by 3 points, due to a significant increase in recall (from 81% to 97%).

Of course, this simple idea works only for entity types with low ambiguity. In fact, it does not improve results if we apply it to NFLGame or ScoreType-*. However, low ambiguity entities are common in many domains (e.g., medical). In such domains, our approach offers a straightforward way to address potential recall errors of a machine learned model.

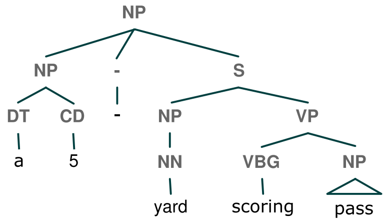
Entity Mentions	P	R	F1
Date	69.5	75.9	72.5
FinalScore	90.9	88.8	89.8
NFLGame	60.5	51.0	55.4
NFLPlayoffGame	37.0	26.3	30.8
NFLTeam	72.4	98.3	83.4
ScoreType-1	69.7	62.1	65.7
ScoreType-2	76.9	63.3	69.4
ScoreType-3	64.3	50.0	56.3
ScoreType-4	72.7	57.1	64.0
Total	73.2	79.2	76.1

Relation Mentions	P	R	F1
fieldGoalPartialCount	81.2	55.4	65.9
gameDate	93.9	27.0	41.9
gameLoser	51.2	17.7	26.3
gameWinner	50.0	8.9	15.2
teamFinalScore	96.5	47.4	63.6
teamInGame	48.3	33.5	39.5
teamScoringAll	86.7	72.9	79.2
touchDownPartialCount	89.1	61.2	72.6
Total	78.5	45.9	57.9

Table 6: Performance with the improved syntactic head identification rules.

5.2 Improving Head Identification for Entity Mentions

Table 1 indicates that most RMD features (e.g., lexical information on arguments, dependency paths between arguments) depend on the syntactic heads of entity mentions. This observation applies to other natural language processing (NLP) tasks as well, e.g., semantic role labeling or coreference resolution (Gildea and Jurafsky, 2002; Haghighi and Klein, 2009). It is thus crucial that syntactic heads of mentions be correctly identified. Originally we employed a common heuristic: we first try to find a constituent with the exact same span as the given entity mention in the parse tree of the entire sentence, and extract its head. If no such constituent exists, we parse only the text corresponding to the mention and return the head of the generated tree (Haghighi and Klein, 2009). Here we argue that the last step of this heuristic is flawed: since most parsers are heavily context dependent, they are likely to not parse correctly arbitrarily short text fragments. For example, the Stanford parser generates the incorrect parse tree:



The syntactic head is "5" for the mention "a 5-yard scoring pass" instead of "pass."[10] This problem is exacerbated out of domain, where the parse tree of the entire sentence is likely to be incorrect, which will often trigger the parsing of the isolated mention text. For example, in the NFL domain, more than 25% of entity mentions cannot be matched to a constituent in the parse tree of the corresponding sentence.

[10] We tokenize around dashes in this domain because scores are often dash separated. However, this mention is incorrectly parsed even when "5-yard" is a single token.

```
teamFinalScore(G, S) :-  teamInGame(T, G), teamScoringAll(T, S).
teamFinalScore(G, S) :-  gameWinner(T, G), teamScoringAll(T, S).
teamFinalScore(G, S) :-  gameLoser(T, G), teamScoringAll(T, S).
    teamInGame(G, T) :-  teamScoringAll(T, S), teamFinalScore(G, S).
   gameWinner(G, T1) :-  teamInGame(G, T1), teamInGame(G, T2),
                         teamFinalScore(G, S1), teamFinalScore(G, S2),
                         teamScoringAll(T1, S1), teamScoringAll(T2, S2),
                         greaterThan(S1, S2).
    gameLoser(G, T1) :-  teamInGame(G, T1), teamInGame(G, T2),
                         teamFinalScore(G, S1), teamFinalScore(G, S2),
                         teamScoringAll(T1, S1), teamScoringAll(T2, S2),
                         lessThan(S1, S2).
```

Table 7: Deterministic inference rules for the NFL domain as first-order Horn clauses. `G`, `T`, and `S` indicate game, team, and score variables.

In this work, we propose several simple heuristics that improve the parsing of isolated mention texts:

- We append "It was " to the beginning of the text to be parsed. Since entity mentions are noun phrases (NP), the new text is guaranteed to be a coherent sentence. A similar heuristic was used by Moldovan and Rus for the parsing of WordNet glosses (2001).
- Because dashes are uncommon in the Penn Treebank, we remove them from the text before parsing.
- We guide the Stanford parser such that the final tree contains a constituent with the same span as the mention text.[11]

After implementing these heuristics, the Stanford parser correctly parses the mention in the above example as a NP headed by "pass". Table 6 lists the overall extraction scores after deploying these heuristics. The table shows that the RMD F1 score is a considerable 4.7 points higher than before this change (Table 5).

5.3 Deterministic Inference for RMD

Figure 1 underlines the fact that relations in the NFL domain are highly inter-dependent. This is a common occurrence in many extraction tasks and domains (Poon and Vanderwende, 2010; Carlson et al., 2010). The typical way to address these situations is to jointly model these relations, e.g., using Markov logic networks (MLN) (Poon and Vanderwende, 2010). However, this implies a complete redesign of the corresponding IE system, which would essentially ignore all the effort behind existing pipeline systems.

Relation Mentions	P	R	F1
fieldGoalPartialCount	81.2	55.4	65.9
gameDate	93.9	27.0	41.9
gameLoser	45.9	27.4	34.3
gameWinner	45.6	25.2	32.5
teamFinalScore	96.5	47.4	63.6
teamInGame	48.1	44.7	46.4
teamScoringAll	86.7	72.9	79.2
touchDownPartialCount	89.1	61.2	72.6
Total	74.2	49.6	59.5

Table 8: Performance after adding deterministic inference. The EMD scores are not affected by this change, so they are not listed here.

In this work, we propose a simple method that captures some of the joint domain structure independently of the IE architecture and the EMD and RMD models. We add a deterministic inference component that generates new relation mentions based on the data already extracted by the pipeline model. Table 7 lists the rules of this inference component that were developed for the NFL domain. These rules are domain-dependent, but they are quite simple: the first four rules implement transitive-closure rules for relation mentions centered around the same NFL-GAME mention; the last two add domain knowledge that is not captured by the text extractors, e.g., the game winner is the team with the higher score. Table 8, which lists the RMD scores after inference, indicates that the inference component is responsible for an increase of approximately 2 F1 points, caused by a recall boost of approximately 4%.

Table 9 lists the results of a post-hoc experiment, where we removed several relation types from the RMD classifier (the ones predicted with poor performance) and let the deterministic inference component generate them instead. This experiment shows

[11] This is supported by the parser API.

	Without Inference			With Inference		
	P	R	F1	P	R	F1
Skip gameWinner, gameLoser	**78.6**	45.6	57.7	**75.1**	48.4	58.8
Skip teamInGame	77.0	43.6	55.7	71.7	49.4	58.5
Skip teamInGame, teamFinalScore	74.5	37.1	49.6	70.9	47.6	56.9
Skip nothing	78.5	**45.9**	**57.9**	74.2	**49.6**	**59.5**

Table 9: Analysis of different combination strategies between the RMD classifier and inference: the RMD model skips the relation types listed in the first column; the inference component generates all relation types. The other columns show relation mention scores under the various configurations.

	EMD	RMD
	F1	F1
Baseline	73.7	49.7
+ gazetteer features	74.0	50.2
+ rule-based model for NFLTeam	75.5	53.2
+ improved head identification	**76.1**	57.9
+ inference	**76.1**	**59.5**

Table 10: Summary of domain customization results.

that inference helps in all configurations, and, most importantly, it is robust: even though the RMD score without inference decreases by up to 8 F1 points as relations are removed, the score after inference varies by less than 3 F1 points (from 56.9 to 59.5 F1). This proves that deterministic inference is capable of generating relation mentions that are either missed or cannot be modeled by the RMD classifier.

Finally, Table 10 summarizes the experiments presented in this paper. It is clear that, despite their simplicity, all our proposed ideas help. All in all, our contributions yielded an improvement of 9.8 F1 points (approximately 20% relative) over the stock IE system without these changes. Our best IE system was used in a blind evaluation within the Machine Reading project. In this evaluation, systems were required to answer 50 queries similar to the examples in Section 4 and were evaluated on the correctness of the individual facts extracted. Note that this evaluation is more complex than the experiments reported until now, because the corresponding IE system requires additional components, e.g., the normalization of all DATE mentions and event coreference (i.e., are two different game mentions referring to the same real-world game?). For this evaluation, we used an internal script for date normalization and we did not implement event coreference. This system was evaluated at 46.7 F1 (53.7 precision and 41.2 recall), a performance that was approximately 80% of the F1 score obtained by human annotators. This further highlights that strong IE performance can be obtained with simple models.

6 Conclusions

This paper introduces a series of simple ideas that improve the performance of IE systems when they are customized to new domains. We evaluated our contributions on a sports domain (NFL game summaries) that is significantly different from the domains used to develop our IE system or the language processors used by our system.

Our analysis revealed several interesting and non-obvious facts. First, we showed that accurate identification of syntactic heads of entity mentions, which has received little attention in IE literature, is crucial for good performance. Second, we showed that a deterministic inference component captures some of the joint domain structure, even when the underlying system follows a pipeline architecture. Lastly, we introduced a simple way to tune precision and recall by combining our entity mention extractor with a rule-based system. Overall, our contributions yielded a 20% improvement in the F1 score for relation mention extraction.

We believe that our contributions are model independent and some, e.g., the better head identification, even task independent. Some of our ideas require domain knowledge, but they are all very simple to implement. We thus expect them to impact other problems as well, e.g., coreference resolution, semantic role labeling.

Acknowledgments

We thank the reviewers for their detailed comments.

This material is based upon work supported by the Air Force Research Laboratory (AFRL) under prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the Air Force Research Laboratory (AFRL).

References

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM)*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*.

Radu Florian, John Pitrelli, Salim Roukos, and Imed Zitouni. 2010. Improving mention detection robustness to noisy input. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3).

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on BioNLP: Shared Task*, pages 1–9. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning (ICML)*.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *EACL*, pages 1–8.

Einat Minkov, Richard C. Wang, Anthony Tomasic, and William W. Cohen. 2006. Ner systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In *Proc. of HLT/NAACL*.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proc. of the Conference of the Association for Computational Linguistics (ACL-IJCNLP)*.

Dan I. Moldovan and Vasile Rus. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies Conference (NAACL-HLT)*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In *Introduction to Statistical Relational Learning*. MIT Press.

Jonathan Schuman and Sabine Bergler. 2006. Postnominal prepositional phrase attachment in proteomics. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 82–89. Association for Computational Linguistics, June.

Mihai Surdeanu and Massimiliano Ciaramita. 2007. Robust information extraction with perceptrons. In *Proceedings of the NIST 2007 Automatic Content Extraction Workshop (ACE07)*.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online qa collections. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*.